

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: David Yach

Art Unit: 2153

Application No.: 09/728,543

Examiner: A. Strange

Filed: 12/01/2000

Attorney Docket No.: 555255012129

For: Virtual Machine Web Browser

REPLY BRIEF

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited today with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on

August 9, 2007

By

Alura Dejean

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Reply Brief is filed in response to the Examiner's Answer mailed June 12, 2007.

I. Status Of Claims

Claims 44-74 are pending and are finally rejected.

II. Grounds Of Rejection To Be Reviewed On Appeal

- Whether claims 44-46, 48-52, 55-57, 58-61, 64-70, 73 and 74 are unpatentable under 35 U.S.C. § 102(e) as being anticipated by Schwartz et al. (US 6,473,609).
- The 102(e) rejections over Lowery (US 6,556,111) and the 103 rejections over Lowery in view of Schwartz, which were fully briefed and argued in the Second Appeal Brief, have now been withdrawn by the Examiner's Answer. Thus, the only remaining ground of rejection to be reviewed is the 102(e) rejection over Schwartz.

III. Argument

A. Schwartz Discloses a Page-Rendering Application

Independent claims 44 and 73 explicitly state that the page-rendered content is browsed “without the use of a page-rendering browser application.” Likewise, claim 64 explicitly states that the web pages are browsed “without the use of a web browser application capable of rendering web pages.” Thus, in order for Schwartz to anticipate these claims, the reference must disclose or suggest the browsing of page-rendered content without an application that renders the content. This is not possible, however, because the Schwartz reference repeatedly and consistently indicates that his invention requires a page-rendering application.

Beginning with the Abstract, Schwartz describes his “interface engine” as something that renders a data file: “The interface engine typically performs tasks that do not require considerable computing power and memory, such as receiving input data from users, **and the rendering of the compact data format received from the link server device, . . .**” Later, in the Detailed Description portion of Schwartz, the “interface engine” is further described as a page-rendering application: “in a method of the present invention, the link server sends over the wireless network a compact data file it generates, and **the interface engine renders the compact data file to cause the display screen to display contents represented by the compact data file.**” (Schwartz, col. 3, ll. 55-60) Later in the Detailed Description, Schwartz further describes his preferred embodiment of the “interface engine,” again revealing that it is a page-rendering application: “According to one embodiment, the SDD file is a group of Imp data. . . There are a set of rules or grammars in the Imp data that an interface engine, **upon rendering of the Imp data**, causes a screen to display the contents of the corresponding markup language file.”

In fact, throughout the Schwartz reference the “interface engine” is repeatedly described as something that renders a data file: “[t]ypically the screen commands are expressed in a form of screen description data (SDD) **that is rendered** in an interface engine in mobile device 350” (col. 9, ll. 36-40); “Upon receiving and **rendering the screen description data**, the interface engine 616 causes display screen 618 to display the information embedded in the screen description data.” (col. 11, ll. 44-47); “. . . while the interface engine in the terminal is only responsible for **rendering the screen description data** to cause the display screen to display contents and receive inputs from a user.” (col. 12, ll. 52-56)

In view of this evidence, there is no real dispute regarding the type of browser application employed in Schwartz – it is clearly a page-rendering application. In fact, in the office action dated September 15, 2005, the Examiner even stated this point, asserting that “Schwartz clearly discloses that the SDD data may be directly rendered by the interface engine of the client device, . . .” Thus, even the Examiner has recognized that the “interface engine” in Schwartz is a rendering application.

There is simply no disclosure or suggestion in Schwartz of browsing content without the use of a page-rendering application. As this limitation is explicitly set forth in each of claims 44, 64 and 73, the anticipation rejection over Schwartz cannot stand and thus the rejection should be reversed.

B. Schwartz Discloses a Data File that is Rendered

The difference between “rendering” a data file and “executing” a program is an important issue in this appeal. The Examiner’s Answer essentially equates these operations, finding that Schwartz’s page-rendering interface engine, which operates on a data file, is the same as the claimed functionality of executing or interpreting an executable program that has been translated from a page-rendered data file into a programmatic language. These operations, however, are not the same, and therefore the anticipation rejection over Schwartz should be reversed.

In each of independent claims 44, 55, 65, 73 and 74, the page-rendered content is translated into a programmatic language that is capable of being executed. In claim 74, for example, an HTML data file, which is a form of page-rendered content, is translated into a JAVA program that is capable of being executed by a JAVA virtual-machine engine at the mobile device. In Schwartz, by distinction, the page-rendered content is not translated into a programmatic language, but instead it is converted into a compacted data file referred to as Screen Description Data (SDD). Moreover, this SDD format, which Schwartz consistently and repeatedly refers to as a “data file,” not an “executable program,” is then undisputedly “rendered” by Schwartz’s “interface engine,” it is not executed.

Schwartz’s SDD form is not an executable program, it is a file of data. The following quotations from Schwartz clearly demonstrate and support this point: “in a method of the present invention, the link server sends over the wireless network **a compact data file it generates**. . .to cause the display screen to display contents represented by **the compact data file**,” (col. 3, ll. 55-60); “According to one embodiment, **the SDD file is a group of Imp data**. . .” (col. 9, line 60 through col. 10, line 6); “[t]ypically the screen commands are **expressed in a form of screen description data (SDD)** that is rendered in an interface engine.” (col. 9, ll. 36-40); “Upon

receiving and rendering **the screen description data**, the interface engine 616 causes display screen 618 to display the information embedded in the screen description data;” (col. 11, ll. 44-47). Thus, as these portions of Schwartz clearly demonstrate, the SDD form is not an executable program, it is a data file.

Further evidence that Schwartz does not disclose the claimed operations of translating the page-rendered content into a programmatic language for subsequent execution is the undisputed fact that Schwartz teaches a page-rendering application at the mobile device — his so-called “interface engine.” As demonstrated above, there is no real dispute that Schwartz’s “interface engine” is in fact a page-rendering application. The reason that Schwartz requires such a page-rendering application is because the SDD form is not executable, but rather it is a data file.

The Examiner’s Answer relies upon the Microsoft Computer Dictionary in support of the assertion that Schwartz’s Screen Description Data is an executable program, but the same reference clearly differentiates between an executable program and something that is rendered:

executable program: “A program that can be run. The term usually applies to a compiled program translated into machine code in a format that can be loaded into memory and run by a computer’s processor. In interpreter languages, an executable program can be source code in the proper format.” Microsoft Computer Dictionary, Third Edition, p. 182.

render: “To produce a graphic image from a data file on an output device such as a video display or printer.” Microsoft Computer Dictionary, Third Edition, p. 406.

These definitions are entirely consistent with the arguments previously presented in this appeal. Schwartz discloses a page-rendering application, the "interface engine," which renders the SDD data file. By distinction the claims of the present invention do not require a page-rendering application, but instead generate an executable program in a programmatic language that is subsequently executed, not rendered, in order to browse the page-rendered content. Based on this additional distinction, the anticipation rejection over Schwartz should be reversed.

Respectfully submitted,

Date: 8/7/07

By: David B. Cochran

David B. Cochran, Reg. No. 39,142
Jones Day
North Point
901 Lakeside Ave.
Cleveland, Ohio 44114
(216) 586-1162